COOLING DEVICE MODELING
WITH COMPUTATIONAL INTELLIGENT

By: Kwok Wai Lun

Supervisor: Prof. Chen Chun

Final Year Project Report -

A report submitted in partial fulfillment of the requirements for

the degree of Bachelor of Engineering in

the Department of Automation and Computer-Aided Engineering

The Chinese University of Hong Kong

May 2020

ABSTRACT

Energy optimization of heating, ventilation, and air conditioning (HVAC) is prominent, especially in Hong Kong, this high-density city has 15% of energy ends up in HVAC. Moreover, with the raise of cheap computational power, machine learning is recognized as the fourth industrial revolution.

This final year project aims to apply machine learning on modeling and controlling a self-made cooling device to optimize the energy usage. The trained indoor environmental predictive model could predict the temperatures and humidity with mean absolute error ranging from 0.5 to 1 depends on the position of sensors. The model is constructed without extra cooling load, the model with cooling load prediction has not been successfully developed due to various factors but the idea will be introduced.

As the no load model has an acceptable low error on temperature predicting, it is capable to predict the indoor temperature distributions for a long future. The best path could be found by predicting the future with all possible control step.

ACKNOWLEDGEMENT

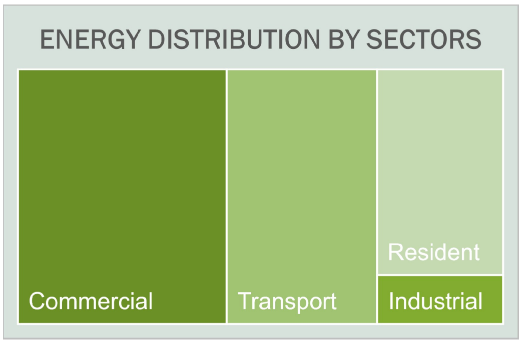
I would like to express my special thanks of gratitude to Prof. Chen Chun for his supervision and support in this project especially allowing me to propose a risky topic that both of us are not familiar with and his encouragement when I was thinking of giving up.

I would also like to extend my gratitude to Mr. Check Chi Ming for technical support on mechanical electronic design.

TABLE OF CONTENTS

MOTIVATION

ENERGY CONSUMPTION ON AIR HANDLING



Hong Kong, a small city with a high population density but little flat land. Suitable lands for large scale renewable energy power plants rare. Yet, modern buildings are everywhere. This uniqueness means that Hong Kong has a lot of potential for energy optimization. According to the energy end-use report 2018, the commercial building sector occupied 43% of the total energy used in this small town which is twice the residential usage and three times higher than the industrial usage. [1]

Figure 1

"Air conditioning" contributes to the highest energy consumption in both commercial and residential sectors, which is 24% and 25% respectively. Air-conditioning has occupied around 15% accumulatively of the total energy used in Hong Kong. Therefore, as an energy and environmental engineering student, I am interested in how we could optimize the usage in this scenario.

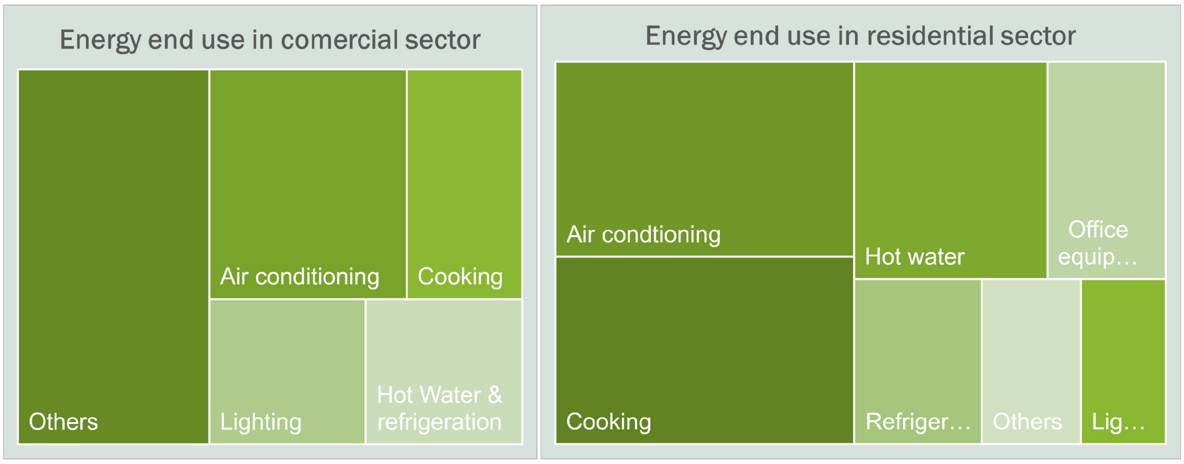


Figure 2

PROBLEM OF HVAC SYSTEM

HVAC, heating ventilation, and air conditioning is a widely used technology to provide indoor thermal comfort and proper air quality. Unlike other building services such as lighting, water supply, and elevators, HVAC could not be defined by the simple physical model. Even an HVAC system obeys the law of physics, the structure of the physical model could be very complex due to the nonlinearity, time-lagging, thermal inertia, and disturbance factors. [2] Although the modern-day HVAC modeling and controlling strategies are mature, there is a lot of room to improve. A traditional HVAC modeling or controlling strategy is designed based on some white-box or mathematical models, with engineering analysis. On the contrary, the black-box model which adopts trial and error to find out the best-fit parameters for the system is a novel solution to the HVAC system. A good white model design requires a good understanding of the HVAC and how it is related to the environment. While comparing to a white-box model, a black-block model is a much simpler solution on HVAC modeling or controlling since it is hard to formulate the complicated HVAC operating environment manually

SCOPE

Machine learning is not skyrocketing technology in computer science, it has been studied for decades. Computational power was expensive and inefficient in the old days but now, it is accessible and very cheap. Which is why machine learning become popular in recent years. According to research the logarithm of millions of instructions per second per price has a 77% growth per year. [3] Modeling and controlling an HVAC system with computational intelligence is an alternative to the traditional methodology. The scope of this project is applying machine learning on a DIY simple cooling device. The scope can be divided into three-part.

1. Data mining
2. Cooling device modeling
3. Optimal control

BACKGROUND AND LITERATURE REVIEW

HVAC MODELING/CONTROLLING METHODOLOGY REVIEW [2]

WHITE BOX (PHYSICS-BASED)

A white box is a mathematical model based on a detail system analysis, which is the most popular and conventional modelling method. A white box is a good method to describe and understanding the behavior of a system, but the HVAC system behaves non-linearly with uncertainty. While modelling an HVAC system, a lot must be considered. For example, material, ventilation, weather and such. Thus, the white box method is very complicated. Heat conduction equation model

- Response factor method
- Heat balance
- Mass and energy balance
- Model predictive control

BLACK BOX (DATA-DRIVEN)

A black box is a statistics tool that finds out the correlation between the input and output without knowing the internal work. It can be done with machine learning. In the past, it is not very efficient and reliable due to limited computational resources. The black box model is simpler but needs regularly update to ensure the performance.

- Unsteady-state model
- Box-Jenkins
- Autoregressive with external inputs
- Autoregressive moving average exogenous
- Output error models
- Neural network-based nonlinear autoregressive model with external input
- Optimal brain surgeon

GREY BOX

A grey Box is utilize the advantages of the white box and the black box. In short, improve the white box model with a black-box model.

DEEP REINFORCEMENT LEARNING FOR BUILDING HVAC CONTROL [4]

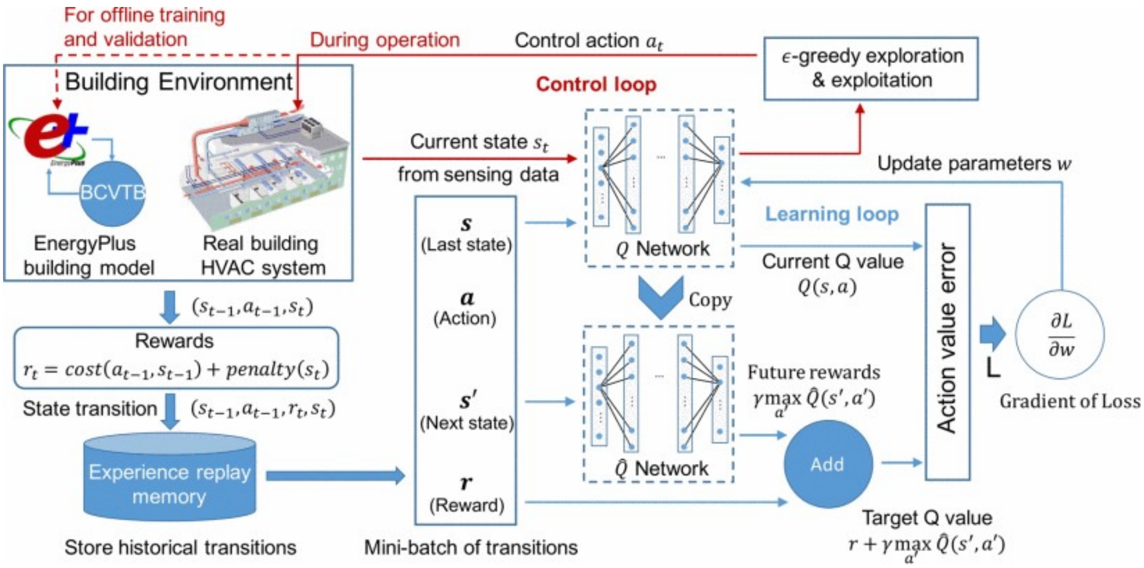


Figure 3 [4]

This research is conducted to design a deep reinforcement learning (RL) with on-off and five-level flow control based on HVAC framework and evaluate its' energy reduction, an unsupervised learning methodology. 20-70% energy saving was observed. The optimal goal is to reduce the energy cost as lower as possible while maintaining a comfortable air condition. The model was trained based on the Energy Plus model. If the optimal goal is achieved, the neural network will be rewarded. The model will tend to gain as much reward as possible by trying out different actions.

APPROACH TO PROBLEM – MACHINE LEARNING

BRIEF

A standard definition of machine learning was given by Tom Mitchell [5], “A computer program is said to learn from experience **E** for some class of tasks **T** and performance measure **P**, if its performance at tasks **T**, as measured by **P**, improves with experience **E.**” To implement a good machine learning, a set of high quality is required as experience, so the first stage of this project is data mining. The second stage is carrying out task T, in this case, it is system modelling/ temperature prediction. The third stage is utilizing the trained model to find out the control strategy for the budget.

FIRST STAGE - DATA MINING (EXPERIENCE)

Machine learning is a powerful tool in every engineering scenarios. However, without an appropriate dataset, it is impossible to build a reliable model. Retrieving desired data from a commercial AC is not feasible. Room for customization is required to obtain an all-rounded dataset. Besides, to reduce the complexity of this project due to the limited budgets and personal experience, a simple cooling device was built with a thermal electric cooler rather than a compressor.

Other than the device, a data-collecting algorithm must be designed to ensure the data is not biased. Data preprocessing was carried out before the data is ready for training.

SECOND STAGE - NEURAL NETWORK AND COOLING SYSTEM MODELING (TASK)

At the very beginning, a machine does not have any information about the system. If it is forced to learn the most energy-efficiency control strategy, the best means is trial and error. (Reinforcement learning) Though, it is time-consuming because the machine may take hundreds of thousands step to reach the best result. Therefore, a virtual characteristic model is useful. If a virtual model is accurate, one could apply any training method on it.

The supervised NN model was introduced for the system modelling instead of a regression model. Even a high degree multinomial regression model could represent the non-linearity of the system, a neural network could work better with a large and comprehensive dataset because it is good at handling intuitive information hide inside the data.

THIRD STAGE – ENERGY OPTIMIZATION

As discussed, reinforcement learning could be useful for decision making. Nevertheless, it was not used in this project. As time is limited, I applied another “hard coding way” for energy optimization. Given that, the model is accurate, all possible outcome of different actions could be predicted. The problem is the complexity could be very high. By default, best path searching is done by the top-down method. For t step predictions, the number of predictions will be. As you can see it is inefficiency and I would apply a method that can greatly reduce the time complexity of searching the best path.

DESIGN DETAIL

COOLING DEVICE

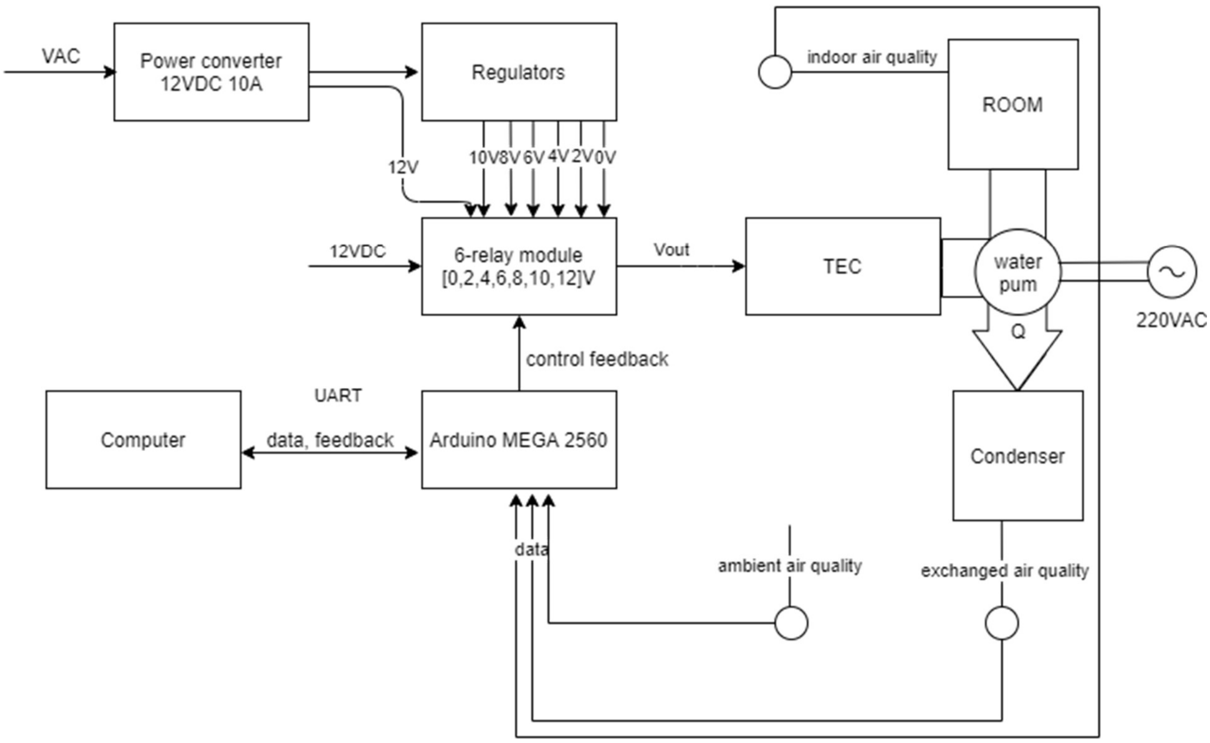


Figure 4

DESCRIPTION

A 12V Thermal electric cooler (TEC1-12712) was used as the cooling unit, the heat side of TEC is connected to a water-cooling device. Heated water will be pumped and run through a condenser (heat exchanger). The TEC has six controllable voltage (0V, 2V, 4V ... 12V) by using Arduino and Six-relays module with 5 regulators. Each regulator corresponding to a specific voltage. Arduino is connected to a computer with USB. The computer will receive data from Arduino and return a voltage setpoint. 4 indoor sensors are distributed in four corners inside the “room”, 1 outdoor sensor and 1 exchanged air sensor. All six sensors are DHT22[1] with 2-5% and 2-degree Celsius absolute error for humidity and temperature, respectively.

COMPONENTS

[1] https://learn.adafruit.com/dht

Cooling unit

- Thermal electric cooler (TEC)
- 220V pump
- Heat sink
- Condenser
- 220V 60Hz AC to 12V 20A dc converter (for TEC)
- 220V 60Hz AC to 5V dc converter (for fans)

Arduino control system

- 6-relays module (6-intervals power control) with regulators
- Sensors
- 60s/cycle[2]

SEQUENTIAL DIAGRAM FOR CONTROLLING

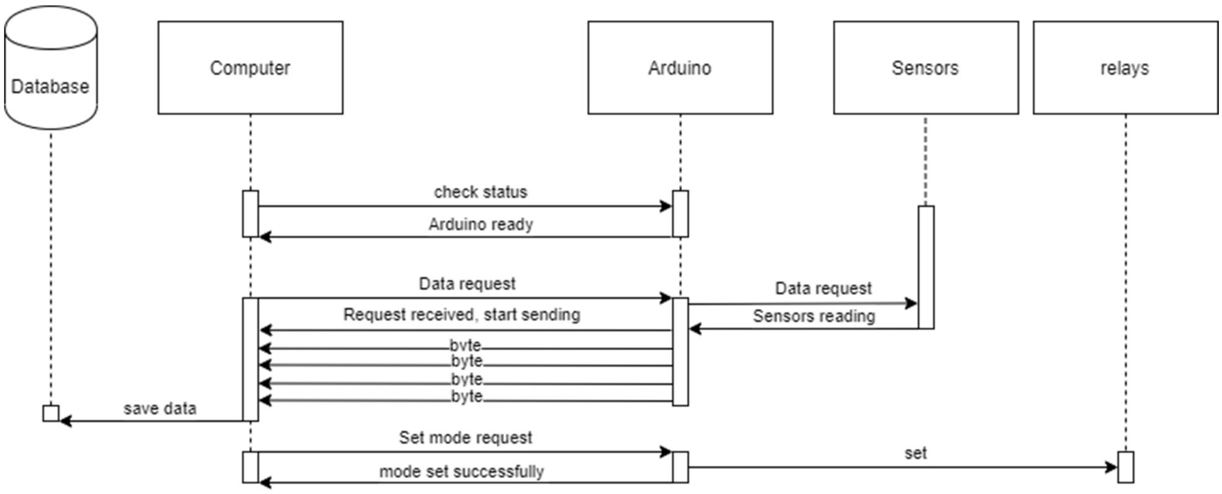


Figure 5

SENSOR PARAMETERS

$$T_e = exchanged\ air\ tempeature,$$

$$H_e = exchanged\ air\ humidity,$$

$$T_o = outdoor\ air\ tempeature,$$

$$H_o = outdoor\ air\ humidity,$$

[2] System will record data and update input voltage once per 60s

$$T_n = indoor\ air\ tempeature, n \in [1,2,3,4]$$

$$H_n = indoor\ air\ humidity, n \in [1,2,3,4]$$

$$V = voltage$$

All the parameters were used as input parameters, and output parameters except voltage which will only be used as input.

DATA COLLECTION ALGORITHM

As mentioned, the quality of data is important to ensure the trained model could be represented in all different situation. In this algorithm, some variable could be customized to ensure I have full control of what kind of data I would like to collect.

First, the algorithm will generate a random number ranging from a to b. If the random number is equal zero, the cooling device will enter a "continuous voltage model", else it will return a random voltage. If a is larger than 0, the system will not enter to "continuous mode" and the voltage will always be completely random. Else if a is zero and the value of b will affect the probability of entering the "continuous mode".

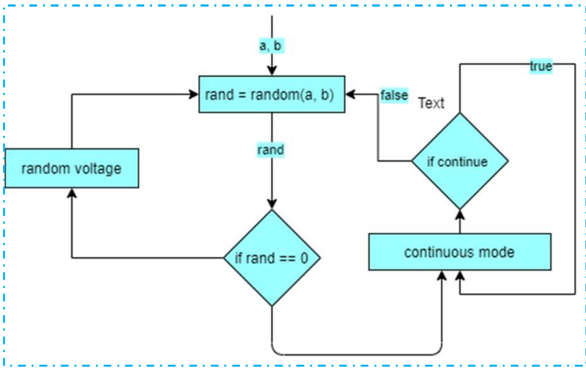


Figure 6

However, a completely random mode will not allow the room temperature to go down so much because the low voltage may only be capable of keeping the temperature instead of decreasing it. Such that, a "continues mode" is designed to avoid the case.

In "continuous mode", duration **D** will first be generated from the range [c, d). A random voltage is then generated. The probability of havening "continuous zero" voltage could be adjusted to prevent the system always remain cold. The voltage will remain unchanged until the duration becomes zero.

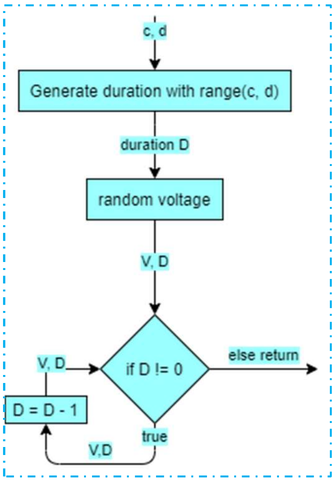


Figure 7

VIRTUAL MODEL (REGRESSION MODEL WITH NEURAL NETWORK)

TRAINING

DESCRIPTION

Neural network is applied in this project. A neural network consists of an input layer, several hidden layers, and an output layer. A neural network with more numbers of hidden layer could help machine to extract more hidden information. On the downside, a deep neural network takes much longer time to reach convergence. In this project, the cooling device is a simple thermal electric cooler with evaporator, the behavior should not be too complicated. So that the shallow neural network technique was used.

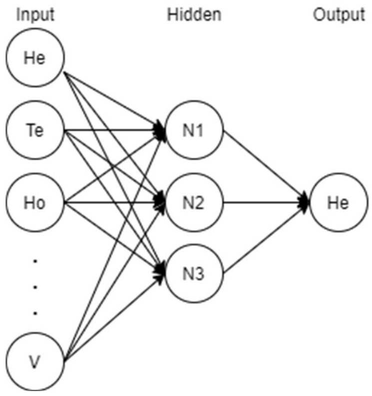


Figure 8

According to the universal approximation theory, a single hidden layer, with enough neurons and nonlinear activation function, is capable to represent any functions. Therefore, single layer and two layers models were applied in the training.

To help neurons understand the data better, data will be preprocessed.

Models with different layers and neurons had been tested and evaluated for the best result.

DATA PREPROCESSING

Neural networks are applied to this project. A neural network consists of an input layer, several hidden layers, and an output layer. A neural network with more numbers of the hidden layer could help the machine to extract more hidden information. On the downside, a deep neural network takes much longer time to reach convergence. In this project, the cooling device is a simple thermal electric cooler with an evaporator, the behavior should not be too complicated. So that the shallow neural network technique.

According to the universal approximation theory, a single hidden layer, with enough neurons and nonlinear activation function, is capable to represent any functions. Therefore, single layer and two layers models were applied in the training.

To help the neuron understand the data better, data will be preprocessed.

Models with different layers and neurons had been tested and evaluated for the best result.:

$$\Delta T_{out,ex} = tmpeature\ difference\ of\ outdoor\ air\ and\ exchanged\ air\ at\ t,$$

$$\Delta T_{ex,t,t-1} = \ exchanged\ airtmpeature\ difference\ of\ of\ t-1\ and\ t\ frames,$$

$$\Delta H_{ex,t,t-1} = exchanged\ air\ humidity\ difference\ of\ t-1\ and\ t\ frames,$$

$$\Delta T_{n,t,t-1} = indoor\ tmpeature\ difference\ of\ t-1\ and\ t\ frames,$$

$$\Delta H_{n,t,t-1} = indoor\ humidity\ difference\ of\ t-1\ and\ t\ frames,$$

$$H_{e,t} = exchanged\ air\ humidity\ at\ time\ t,\ \ T_{e,t} = exchanged\ air\ tempeature\ at\ time\ t,$$

$$H_{out,t} = outdoor\ air\ humidity\ at\ time\ t,\ \ T_{out,t} = outdoor\ air\ tempeature\ at\ time\ t,$$

$$H_{n,t} = indoor\ air\ humidity\ at\ time\ t,\quad T_{n,t} = indoor\ air\ tempeature\ at\ time\ t,$$

$$V_{t-1} = voltage\ at\ t-1,\qquad V_t = voltage\ at\ t,$$

$$for\ n\ \in [1\ ,2\ ,3\ ,4\ ]$$

ACTIVATION FUNCTION

Without an activation function, a neural network could only be a complicated linear model. Therefore, adding in-linearity to the model is important. In this project, relu is used, which emphasizes the positive value from the previous layer and ignore the negative one. Relu is a role player in neural networks. There is no doubt that relu is better than the other traditional activation function. One could consider using advanced relu if relu does not work properly but in this project, relu were working well.

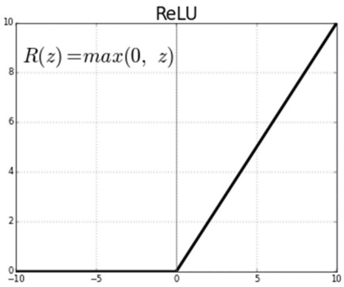


Figure 9

LOSS FUNCTION

The function used for gradient descent means absolute error. The advantages of mean square error would punish large error more. Consider the following case. Given that we have two models with different errors: {0.5, 0.5, 0.5} and {1.2, 0.1, 0.1}. The mean absolute errors are 0.5 and 0.467 respectively, and the mean square errors are 0.25 and 0.487 respectively. With mean square error, the first model will be selected as a better solution, since the second set has a higher chance of seeing a large error of specific item. Vice versa, the mean absolute error only cares about the net average error. In my case, I could tolerate a relatively small error, but not the large error. If the model could constantly predict with small error, the prediction could be used for input again to do a sequential prediction to predict the behavior after certain time steps. If not, large errors will appear more frequently. In this case, if large error prediction is feed into the model again as an input, the output will continue to have large errors, since the input is already misleading. Therefore, means square error will be used as a loss function, the mean absolute error will be used for evaluation.

$$Mean\ square\ error = \frac{\sum_{n}^{N}\left(y_{label} - y_{prediction}\right)^2}{N},$$

$$Mean\ absolute\ error = \frac{\sum_{n}^{N}\left|y_{label} - y_{prediction}\right|}{N}$$

EVALUATE

Models were evaluated with means absolute error. Not only the one frame prediction is evaluated. Predictions after time t will also be evaluated. Data were split into three sets. 60% for training, 20% for validation, 20% for testing. This method is known as cross-validation.

OPTIMIZATION

To find out the best path to reach the desired temperature T within a given time t, all possible path to reach the desired temperature need to be listed out. The permutation with repetition is generated. Compute all elements of permutation are time-consuming. The hack is to apply a “binary search” approach. To illustrate the idea, imagine the desired temperature is 20 degree Celsius and the time constrain is 3 minutes

t=1	t=2	t=3
0V	0V	0V
2V	2V	2V
4V	4V	4V
6V	6V	6V
8V	8V	8V
10V	10V	10V
12V	12V	12V

Since energy is conserved, for a {12V,12V,12V} sequence the temperature must drop more than {10V,10V,10V} or {10V,10V,12V} or {10V,12V,12V}. Three N Voltage sequence should always capable to cool more than the one that has at least one frame which voltage is below N and the rest are equal and not higher than N.

Figure 10

Mathematically,

$$Let\ T(\boldsymbol{X})\ be\ the\ tempeature\ after\ t\ minutes\ for\ \boldsymbol{X} \in\ \mathbb{R}^t$$

$$for\ t = 3,$$

$$Let\ U\ be\ a\ set\ \rightarrow U = \{N, N, N\}\ and$$

$$Let\ V\ be\ a\ set\ \rightarrow\ V = \{M_1, M_2, M_3\}$$

$$If\ all\ M \leq N\ and\ at\ least\ \boldsymbol{one}\ M < N\ then$$

$$T(N) < T(M)\ and\ P(N) > P(M)$$

With this property:

If the desired temperature could not be reached by {12V, 12V, 12V}, all the other path could not either.

If the desired temperature could be reached by {6V, 6V, 6V}, the most energy efficient path will be bounced between {0V, 0V, 0V} and {6V, 6V, 6V}.

A binary search algorithm present searching from the middle of an **ordered** set, if the value is not presented in the middle, split the set into half and search in the half of the half. For example, if we would like to find number 10 in the following set:

Step 1, access the middle:

2	3	4	7	9	10	11	12	14

Since 9 < 10, we preform binary search on the right half:

9	10	11	12	14

Since 11 > 10, we preform binary search on the left half:

9	10	11

It takes 5 comparisons to reach 10, with traditional searching, but only 3 in binary search. When the set is big, the reduction in complexity is much more significant. Therefore, I applied a similar idea on controlling my device.

Here is the energy budget control algorithm:

The time step is limited to 5, if the temperature could not be reached within 5, return a control sequence {12V, 12V, 12V, 12V, 12V, 12V}

Procedure:

Step 1:

Search {6V, 6V, 6V, 6V, 6V} (The middle)

Step 2:

If predicted temperature > desired temperature

Search the right half

Else

Search the left half

Step 3:

repeat until the closet temperature found (constrain predicted temperature <= desired temperature)

Step 4:

Set the firs element of control as the result from last step.

For example, if it is 6V. The first element will be {6V, ?V, ?V, ?V, ?V, ?V}.

Step 5:

Repeat step 1 to 4 for the next elements but this time the maximum will be 6V.

Find best N, {6V, N, N, N, N, N} for N $\in (0,2,4,6)$

If best is 4V,

Find best N, {6V, 4V, N, N, N, N} for N ∈ $(0,2,4)$

The reason the leftmost element would always be the largest is to avoid overshoot, since large voltage tends to over shoot, if large voltage is set at the earliest, our algorithm should be able to avoid it by having small voltage on the rest. On the other hands, the binary searching will always find out the most power efficiency paths because if there exist two path, one consist of a higher voltage and the other one is a combination of lower voltage, the algorithm would always select the after one. As explained above, $if\ T(N) < T(M)\ then\ P(N) > P(M)$. If there exist an N_1 and an N_2, implies $T(N_1\ ) < T(N_2\ )\ the\ algo\ will\ select\ N_1\ since\ P(N_1) < P(N_2)$.

COOLING LOAD PREDICTION

Assume we had already built an accurate no cooling load model. It could be used as a base to build a model with the cooling load. When extra cooling loads are presented, the predicted value would be different from the real measurement. The difference could be collected, and then, to be trained again. This time, the input will be the error between the predicted value and the actual value. The output value is the actual value. The other entities in the old model should be tested if they could help to improve the loos. The logic is shown below.

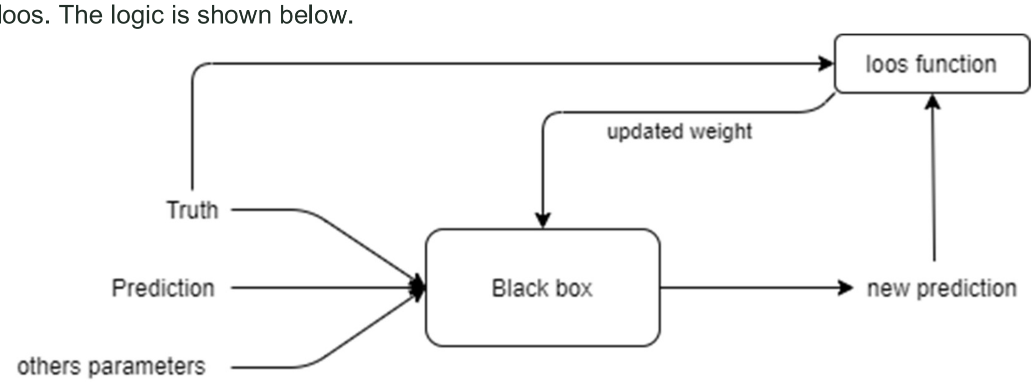


Figure 11

RESULT

RAW DATA PREVIEW

Sample:

Date	H_o	T_o	H_e	T_e	H_1	T_1	H_2	T_2	H_3	T_3	H_4	T_4	V	I
5/4/2020 1:06	82.5	22.9	81.2	25.1	70.3	22.4	80.6	21.9	73.2	21.8	70.9	21.9	10	7.08
5/4/2020 1:07	81.4	23.1	79.2	25.6	63.8	21.7	72.2	21	66.3	20.8	64.5	21	10	6.89
5/4/2020 1:08	80.5	23.4	76.7	26.2	64.7	21.1	73.9	20.1	67.6	20	65.7	20.2	10	6.84
5/4/2020 1:09	79.5	23.6	74.5	26.7	66.1	20.5	76.4	19.4	69.5	19.3	67.5	19.5	10	6.89
5/4/2020 1:10	78.6	23.8	72.6	27.2	67.9	19.9	78.7	18.9	71.4	18.8	69.4	19	10	7.03
5/4/2020 1:11	77.7	24	71.1	27.5	69.3	19.5	80.3	18.6	72.9	18.5	70.8	18.7	10	6.93
5/4/2020 1:12	77	24.1	70	27.7	70.8	19.2	81.8	18.3	74.1	18.2	71.9	18.4	10	6.79
5/4/2020 1:13	76.5	24.3	69.2	27.9	71.6	18.9	82.7	18.1	75.1	18.1	72.7	18.3	10	6.98
5/4/2020 1:14	76.3	24.3	68.6	28.1	72.5	18.8	83.5	18	75.8	17.9	73.4	18.1	10	6.89
5/4/2020 1:15	76.2	24.4	68	28.2	73.3	18.6	84.2	17.8	76.5	17.8	74	18	10	6.98
5/4/2020 1:16	76.1	24.4	67.6	28.4	73.9	18.5	84.9	17.8	77	17.7	74.5	17.9	10	6.89
5/4/2020 1:17	75.8	24.5	67.1	28.5	74.3	18.4	85.3	17.7	77.5	17.6	74.9	17.8	10	6.98
5/4/2020 1:18	75.7	24.5	66.6	28.6	74.6	18.3	85.7	17.6	77.8	17.5	75.2	17.8	10	6.98
5/4/2020 1:19	75.1	24.6	66.1	28.6	75.1	18.3	86	17.6	78.1	17.6	75.4	17.7	10	6.79
5/4/2020 1:20	74.3	24.7	65.9	28.7	74.9	18.2	86.1	17.6	78	17.5	75.4	17.7	10	6.98
							….							

Figure 12

Date	H_o	T_o	H_e	T_e	H_1	T_1	H_2	T_2	H_3	T_3	H_4	T_4	V
MAX	86.1	30.8	91.1	34.5	99.9	27.8	99.9	27.9	99.9	27.9	99.9	28	12
MIN	58.7	19.9	41.9	20.9	54.7	13.8	60.4	13	56.8	12.9	55.7	13.2	0
SD	4.15	1.86	7.91	2.31	11.2	2.66	8.02	2.87	10.2	2.88	10.4	2.83	4.16
MEAN	73.5	24.7	71.3	27.0	85.6	19.7	92.4	19.2	87.8	19.3	85.5	19.4	5.28

Figure 13

TRAINING PROCESS

Four different datasets with different input parameter were tested.

Only one indoor air data was used in alpha, beta and cita, which were tested for the effect of different preprocessing strategies. Followings are a few questions I would like to figure out.

1. Is predicting the change in temperature more accurate?
2. Is inputting the change in last minutes more accurate?
3. Is implementing both 1 and 2 even better?

Data set	Input									Output	
Alpha	$H_{1,t-1}$	$T_{1,t-1}$	V_{t-1}	$H_{out,t}$	$T_{out,t}$	$H_{e,t}$	$T_{e,t}$	$H_{1,t}$	$T_{1,t}$	V_t	$H_{1,t+1}$	$T_{1,t+1}$
Beta	$\Delta H_{1,t,t-1}$	$\Delta T_{1,t,t-1}$	V_{t-1}	$H_{out,t}$	$T_{out,t}$	$H_{e,t}$	$T_{e,t}$	$H_{1,t}$	$T_{1,t}$	V_t	$H_{1,t+1}$	$T_{1,t+1}$
Cita	$\Delta H_{1,t,t-1}$	$\Delta T_{1,t,t-1}$	V_{t-1}	$H_{out,t}$	$T_{out,t}$	$H_{e,t}$	$T_{e,t}$	$H_{1,t}$	$T_{1,t}$	V_t	$\Delta H_{1,t+1,t}$	$\Delta T_{1,t+1,t}$

Figure 14

After the best method was figured out, a new dataset Delta was created, and the delta will contain information about all indoor data. Since, this time not only the model could learn how the air interacts with the past data, but also different position.

Input							
$\Delta H_{out,ex,t}$	$\Delta T_{out,ex,t}$	$\Delta H_{1,t,t-1}$	$\Delta T_{1,t,t-1}$	$\Delta H_{2,t,t-1}$	$\Delta T_{2,t,t-1}$	$\Delta H_{3,t,t-1}$	$\Delta T_{3,t,t-1}$
$\Delta H_{4,t,t-1}$	$\Delta T_{4,t,t-1}$	V_{t-1}	$H_{out,t}$	$T_{out,t}$	$H_{e,t}$	$T_{e,t}$	$H_{1,t}$
$T_{1,t}$	$H_{2,t}$	$T_{2,t}$	$H_{3,t}$	$T_{3,t}$	$H_{4,t}$	$T_{4,t}$	V_t
Output							
$H_{e,t+1}$	$T_{e,t+1}$	$H_{1,t+1}$	$T_{1,t+1}$	$H_{2,t+1}$	$T_{2,t+1}$	$H_{3,t+1}$	$T_{3,t+1}$
$H_{4,t+1}$	$T_{4,t+1}$	V_{t+1}					

Figure 15

TRAINING RESULT

OVERFITTING

One of the most concerned the problem of machine learning is overfitting. When a model is overfitted, it will bias to the training data. According to observation, all trained model has similar error on all training example, validation example and testing example so the difference will not be introduced. One may assume overfitting was not presented in all evaluated models. Moreover, the reason the model never overfit is that the training set and validation set itself does not vary much as the dataset is very large and shuffled.

EFFECT OF DIFFERENT INPUT PARAMETERS PREPROCESSING

A bilayer 36 × 36 NN was used to predict the temperature of indoor position 1 at t+1 all 4 datasets. The result was recorded after 110000 epochs[3].

Dataset	*Mean square root error (MSE)*	*Mean absolute error (MAE)*
Alpha	0.0133	0.0745
Beta	0.0118	0.0696
Cita	0.0132	0.0694

Figure 16

The result shows that inputting the change in air quality could help the prediction. However, predicting the change in temperature does not show an improvement, Instead, the MSE increase in Cita. So that, in the delta the "beta methodology" is adopted.

EFFECT OF DIFFERENT MODEL

Besides, training with different input parameters preprocessing, different models were tested out. In general, a more complicated relationship required a deeper neural network or a single-layer neural network with great number of neurons. Prediction temperature at t+1 with delta was used as the skeleton in this part. The result was recorded after 110000 epochs.

Models	*MSE*	*MAE*
Single-layer 36	0.0087	0.0581
Single-layer 72	0.0086	0.0585
Single-layer 128	0.0085	0.0587
Bi-layer 36 × 36	0.0082	0.0566
Tri-layer 128 × 256 × 128	0.0082	0.0568
Tri-layer 16 × 36 × 16	0.0096	0.0635

Figure 17

It is shown that difference between different models is not much. While a deeper neural network, with fewer neurons 16,36,16 has an error much larger than the one with more neurons 128 × 256 × 128, which may indicate a deeper neural network may require more neuron to ensure the model is complicated enough to be representative.

The point of convergence is around MSE 0.008. A more complicated model does not help so much may because the data itself could only do a job as good as a single-layer 36 model, a more complicated model

[3] An epoch is a term used in machine learning and indicates the number of passes through the entire training dataset the machine learning algorithm has completed.

does not have much meaning since it is already well described by a single-layer network.

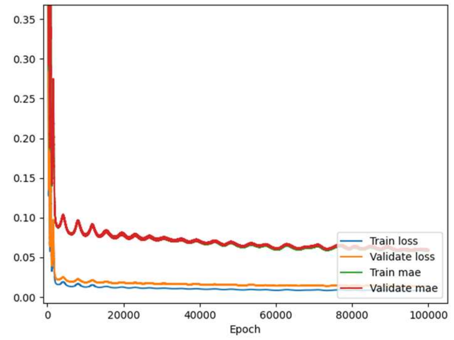


Figure 18: single-layer 36

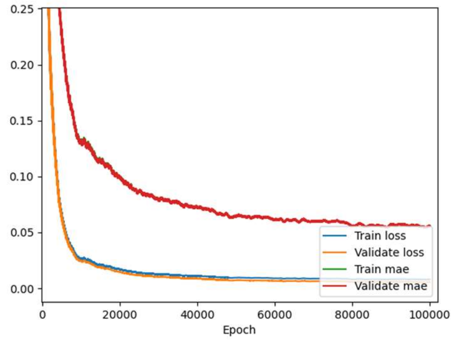


Figure 19: bi-layer 36 ×36

Above is the learning curve of single-layer 36 model and bi-layer 36 ×36 model respectively, (note that two graphs are not in scale) The single layer reach MAE 0.1 much faster than the bi-layer one. Nevertheless, the time used to reach the point of convergence does not vary much.

The analysis process was repeated to predict the humidity. Finally, bi-layers 36 × 36 models are used for temperature prediction and single layer 256 models are used for humidity prediction. The result of all models is listed as follow:

Model	*MSE*	*MAE*
Delta_Hex	0.4155	0.4498
Delta_Tex	0.0176	0.0836
Delta_H1	1.6829	0.8993
Delta_T1	0.0089	0.0615
Delta_H2	1.2065	0.7304
Delta_T2	0.0221	0.0664
Delta_H3	1.3741	0.7371

Delta_T3	0.0198	0.0570
Delta_H4	1.1730	0.6920
Delta_T4	0.0189	0.0542

Figure 20

The rate of change of humidity is much larger than that of temperature. Larger error obtained from humidity is reasonable.

SEQUENTIAL PREDICTION

is given a starting point, the machine could predict the temperature after a long time. In such case, we could set an optimal temperature and try out different voltages setting to find out the best decision. In this part, an example of long-time prediction will be illustrated with comparison with the actual value. The output of prediction will be feed into the model with new voltage as an input. To simplify the illustration, only one indoor air prediction will be shown. Detail could be found in Appendix.

Value from virtual mode[4]		Value from real device	
Humidity	Temperature	Humidity	Temperature
53.86	25.02	57.1	25.1
81.68	24.47	58.8	24.2
69.08	24.23	60.9	23.5
63.51	23.53	62.7	22.8
62.18	22.86	64.6	22.3
63.45	22.24	65.9	21.9
65.86	21.74	66.9	21.6
68.31	21.3	67.5	21.3
70.15	20.93	68.2	21.1
87.02	20.87	82.7	21
92.56	21.22	89.7	21.4
92.88	21.66	90.2	21.7
91.44	22.04	89.5	22.1
90.55	22.36	88.3	22.3
89.51	22.62	87.3	22.6
88.7	22.82	86.5	22.8
88.02	22.98	85.6	22.9
87.48	23.09	85	23

[4] Values were round down to 2 decimal digits

87.06	23.18	84.6	23
86.58	23.24	84.3	23.1
66.75	22.96	63.9	22.9
64.43	22.3	63.3	22.3
91.39	22.07	88.9	22.1
91.63	22.41	90.5	22.3
73.9	22.36	72.3	22.3
67.13	21.96	66.6	22
67.27	21.52	66.6	21.5
68.17	21.13	67.3	21.1
69.87	20.79	68	20.8
71.55	20.51	69.3	20.5
72.84	20.26	70.2	20.2
73.73	20.04	70.9	20
74.39	19.85	71.5	19.9
74.94	19.69	72.1	19.7
75.35	19.55	72.7	19.6
75.7	19.43	73	19.5
76.04	19.32	73.7	19.4
76.35	19.21	73.8	19.4

Figure 21

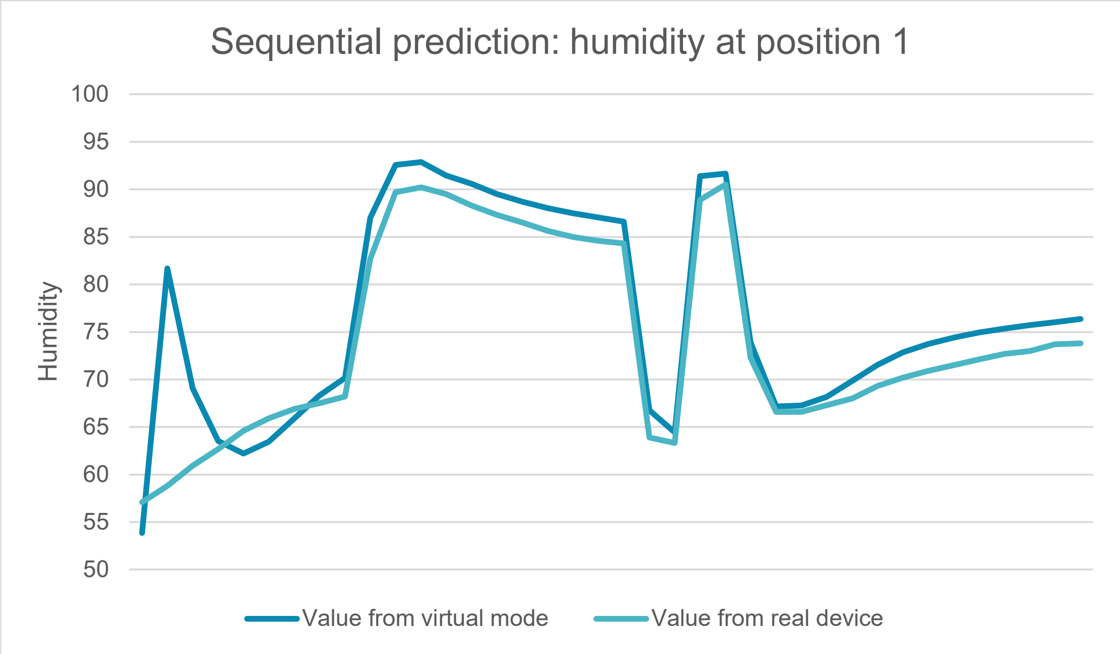


Figure 22

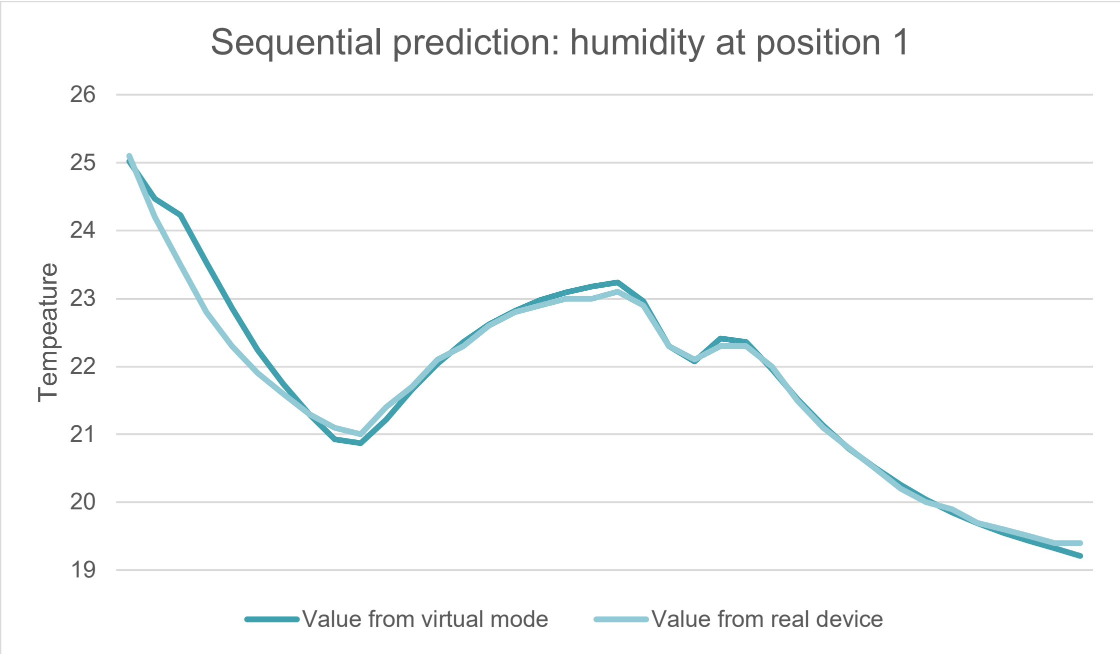


Figure 23

CONCLUSION

A high accuracy model was successfully trained. It is concluded that the accuracy of a model is bounded by the quality of data. In machine learning, how to collect a pool of decent data is much more important than how to design a good model. In general, a neural network could represent any possible function and should not be a big concern if a model could not speak for the data. Instead, answering the question "Does the data itself is comprehensive enough to be define?" is more meaningful.

Even though a cooling load model was not present in this project, I strongly believe that it could be achieved if a no-load model is accurate.

Given that a real HVAC system could be very complicated, it consists of different components. For example, chiller, reheater, heat exchanger, air duct, vents etc. The input parameter required is much more than those in this project. It could be messy to handle such huge amount of data.

Again, computation intelligence would play a big role in the future of energy harvesting industry, we should work harder to bring this to reality.

DIFFICULTY

SUMMARY

Throughout the project I had faced a lot of obstacles, the biggest problem is I had overestimated my capability also underestimated the difficulty of building all those things from starch. Especially, I had zero knowledge of machine learning when the idea of this project came up to my mind. Luckily, I had made use of the entire summary holiday to grab some basic concept of machine learning. However, I after all this long progress, I think I still do not think I have done a very good job on the learning part but I myself is satisfactory with the result.

AT THE VERY BEGINNING

I requested my supervisor, Prof. Chen for a dataset from an actual building. However, when I started playing with those data, I realized if the data is not collected under a well-planned purpose, it is barely meaningful. I had been struggled for weeks to figure out, how should I start this project. After reviewing some research on this scenario, I found out that adopting machine learning in a real HVAC could be complicated. Therefore, I thought about taking an easier way, a thermal electric cooling device. As the system has much fewer components, it should not be too complicated. That is how I started designing my cooling device.

BUILDING THE COOLING DEVICE

I had spent nearly one month to design my first draft and started sourcing and printing my CAD drawing. After another month, I had my first designed come up. Since I was planning to create a cooling device which looks like a real refrigerator. Not only there was a condenser, but also an evaporator. Turns out, the cooling plate is too weak to effectively bring the heat from the indoor to outdoor. Also, the room was too big that the cooling capacity was insufficiency. And that is why I have my latest design now. The plan was remade a smaller one with totally the same material. Unfortunately, the laser cutter was burned and I replaced my design with PP foam. In this stage, the part bugging me the most was sourcing. I was not very familiar with electronic circuit design, I found something that I had missed something in my design. Also, because of the continuous protests in the first semester my build had been delayed and was finally finished at holiday between semesters.

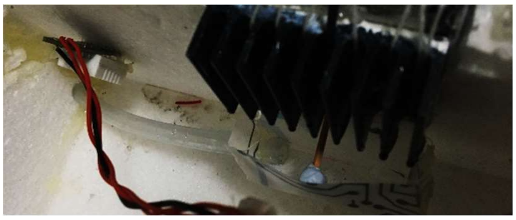
START COLLECTING DATA

This part is the most difficult part of the project. Not only I had to design an algorithm that could be comprehensive enough for training, coding a protocol to allow computer and Arduino to communicate was not simple. Yet, the best solution is always to work harder and non-stop debugging.

For the collection part, I designed an algorithm that I can have full control of how likely the specific mode will occur and every time if it occurs, how long would it last by simply changing and it worked quite well.

Other than the software part, the sensors are also fragile. The jump wires become loose after times. I had performed not less than 10 times of electronic debugging to figure out which wire was loosened. Eventually, I stuck everything tight with some UHU glue. Even it is not very “engineering”, it was the only tools I have in my home during the COVID outbreak.

I also realized the problem of dripping water after the first time I run the device for days. I found out that the room was full of water and I must determine how to make a drainage system. Again, I used some UGU glue with a plastic folder and pipe to make it happen.



TRAINING

The most challenging part of challenging was its time complexity. Since the dataset is a bit big so that the time required to go through entire dataset once is quite long. Every model has to be trained for at least 20 minutes until convergence. Every time I tested out a model and wanted to try a new one, it took me hours.

DHT-22

DHT-22, the temperature and humidity used in this project are accurate but not reliable. The humidity of the sensors would frustrate when the temperature changed, it could be the reason for humidity prediction is not as accurate as temperature. I once was afraid that it could be troublesome to deal with. However, the impact on humidity is relatively low. A few degrees error would not affect the prediction of temperature much. The model was also improved, to help to increase the humidity prediction accuracy. At the very first moment, the MAE was around 1.5 and after a lot trail, it is now around 0.7.

THE SCOPE IS TOO AMBIGUOUS

In the origin planning, I supposed to train another different model which could handle cooling load prediction. However, due to the outbreak of COVID-19, I could not finish the cooling load generator at my home and left that plan behind. I also planned to use a machine learning approach to do the energy optimization, but the time was running out. I realized setting an achievable objective is also important on project management.

RREFERENCE

[1] Electical and Mechanical Services Department, "Hong Kong Energy End-use Data 2018," 9 2018. [Online]. Available: https://www.emsd.gov.hk/filemanager/en/content_762/HKEEUD2018.pdf.

[2] R. Z. Homod, “Review on the HVAC System Modeling Types and the Shortcomings of Their Application,” *Journal of Energy,* 9 February 2013.

[3] H. Koh and C. L. Magee, "A functional approach for studying technological progress:," *Technological Forecasting & Social Change,* 1 6 2006.

[4] Tianshu Wei ; Yanzhi Wang ; Qi Zhu, "Deep reinforcement learning for building HVAC control," in *2017 54th ACM/EDAC/IEEE Design Automation Conference*, 2017 .

[5] T. Mitchell, Machine Learning, New York: McGraw-Hill Science/Engineering/Math, 1997.

[6] The Economist, "Drastic falls in cost are powering another computer revolution," 2019.

APPENDIX

THERMAL ELECTRIC COOLER (TEC1-12712)

MAX. delta T. = 62 Degree C.

MAX. current = 12A

Rated voltage = 12VDC

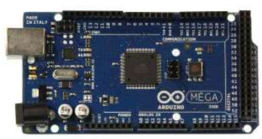
MAX voltage = 15.5V

Cooling power = 114W

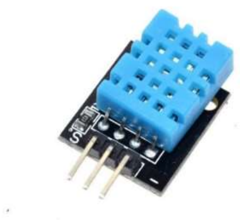
SIX-RELAYS MODULE

HIGH CURRENT REGULATORS

ARDUINO MEGA



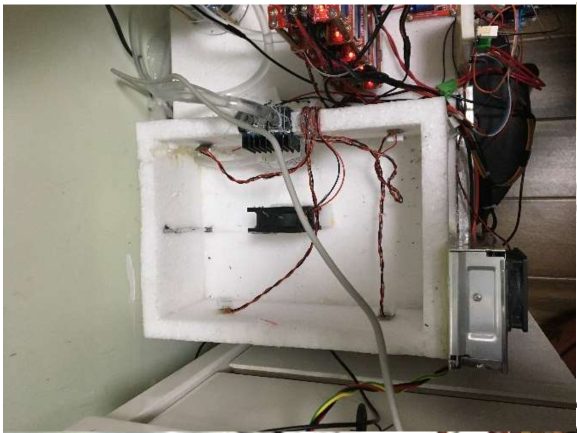
DHT-22(HUMIDITY AND TEMPERATURE SENSOR)

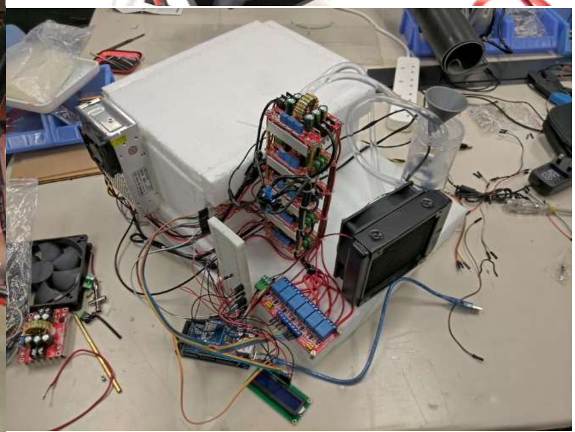
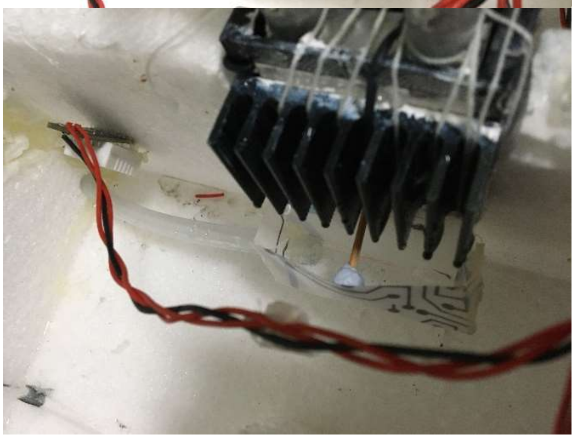


220V WATER PUMP

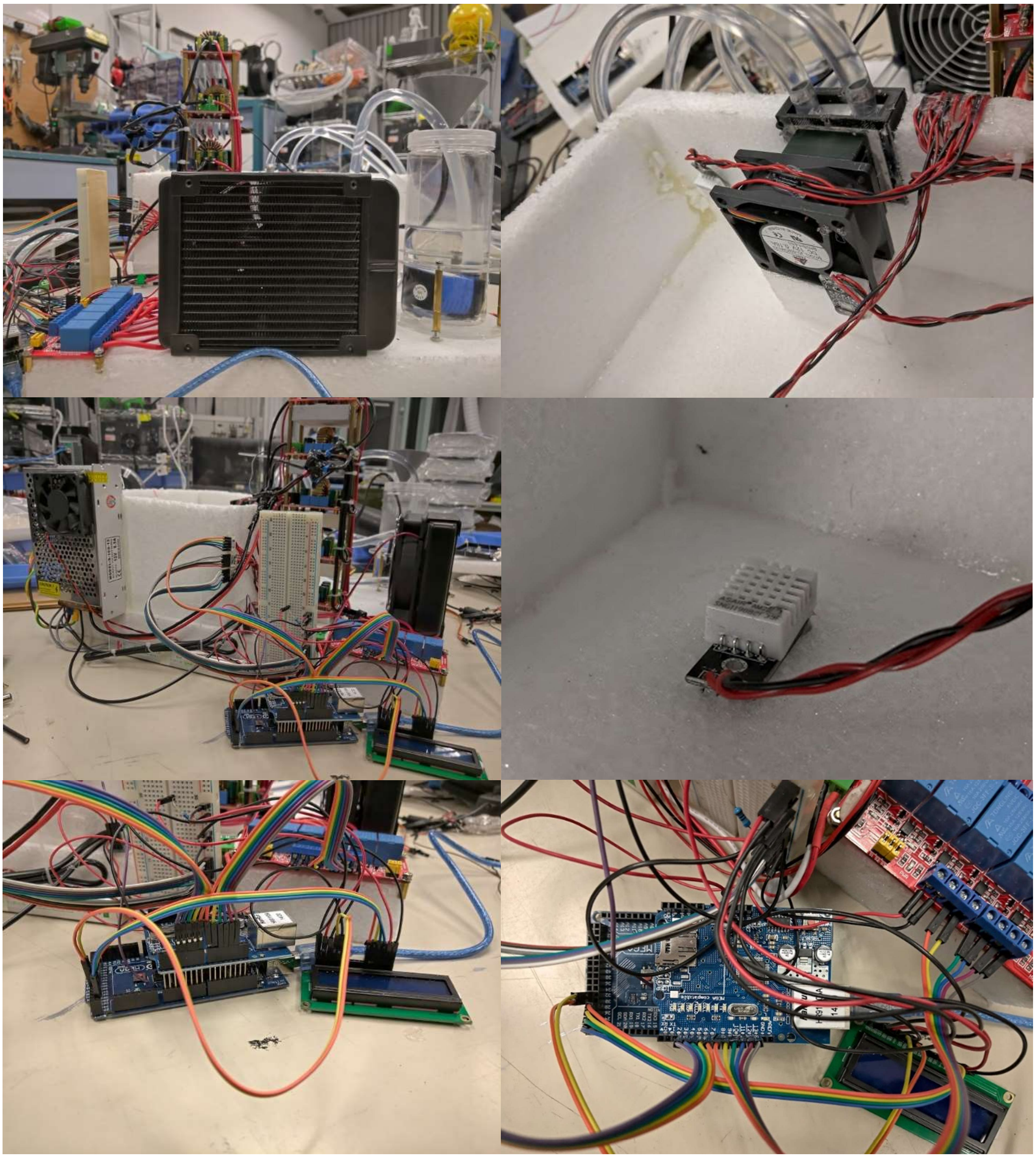
220V REGULATOR

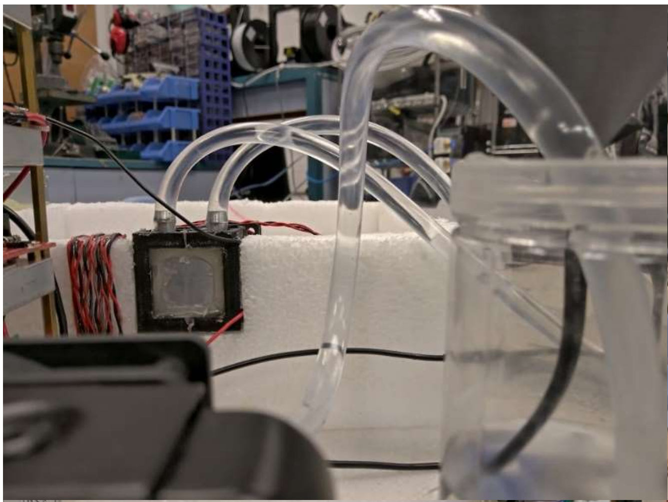
LATEST SETUP

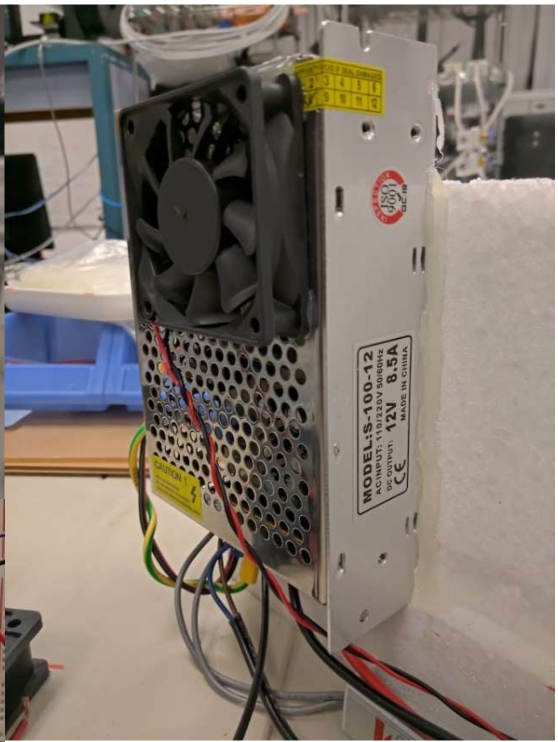


SECOND PROTOTYPE

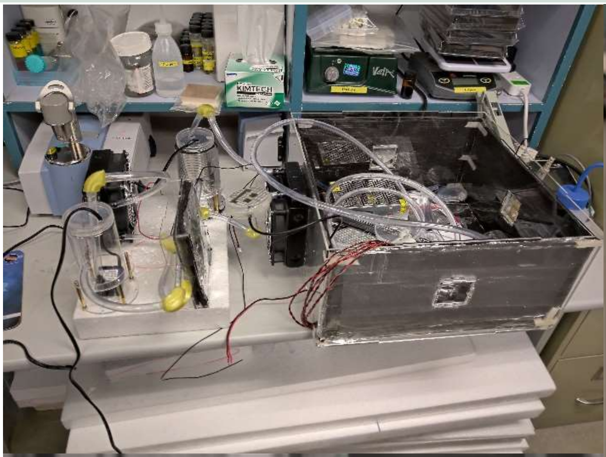


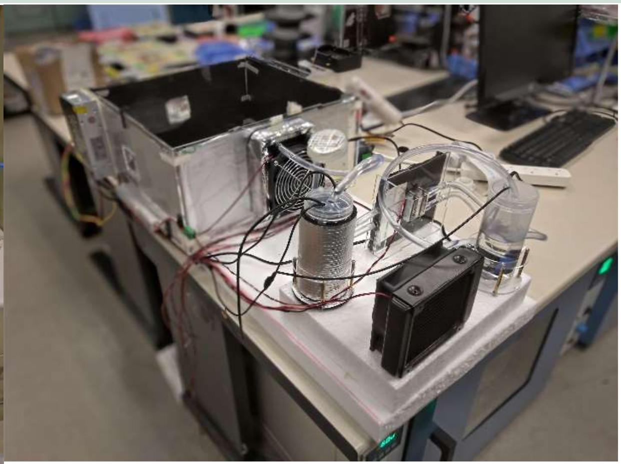


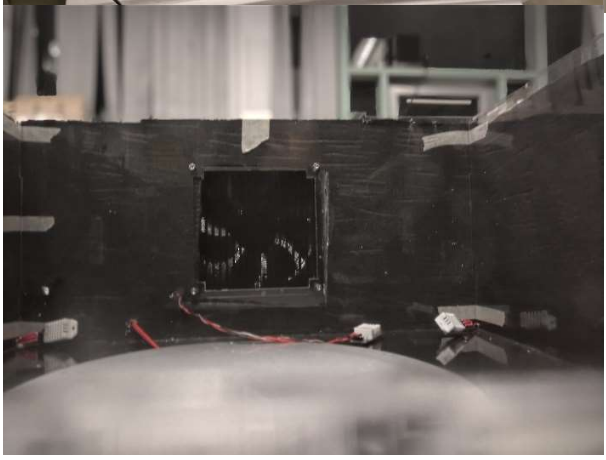

MODEL:S-100-12
AC INPUT: 110/220V 50/60Hz
DC OUTPUT: 12V 8.5A
MADE IN CHINA
CAUTION


AC200-240V
DC 12V
1.5A
18W

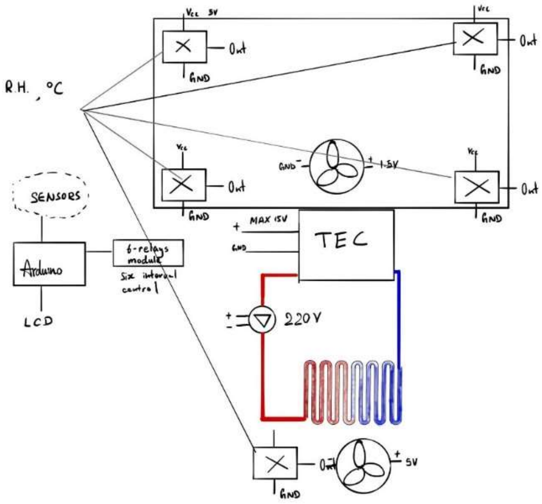
FIRST PROTOTYPE



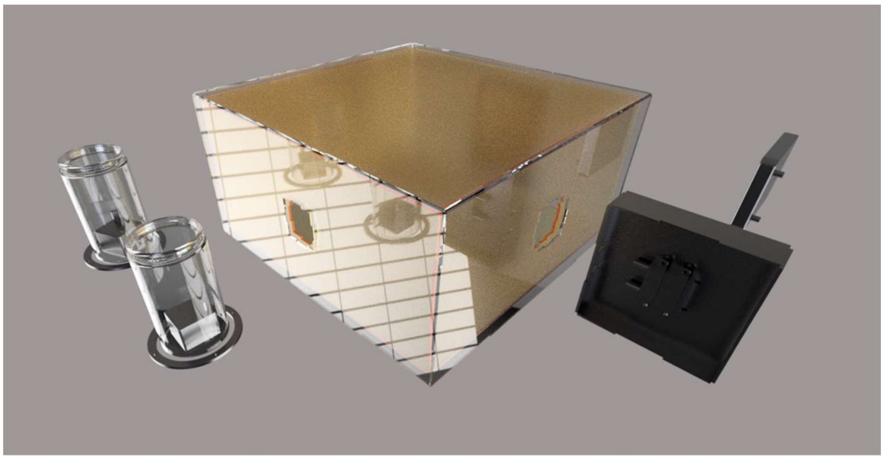




DESIGN SKETCH



FUSION CAD DRAWING RENDER



LEARNING CURVE OF MODELS

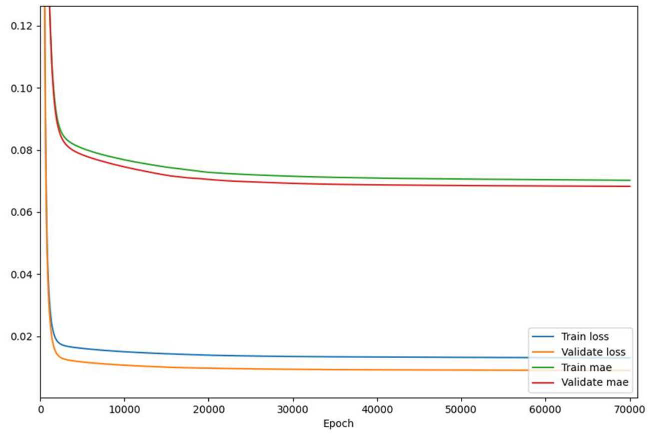


Figure 24: beta single layer 36

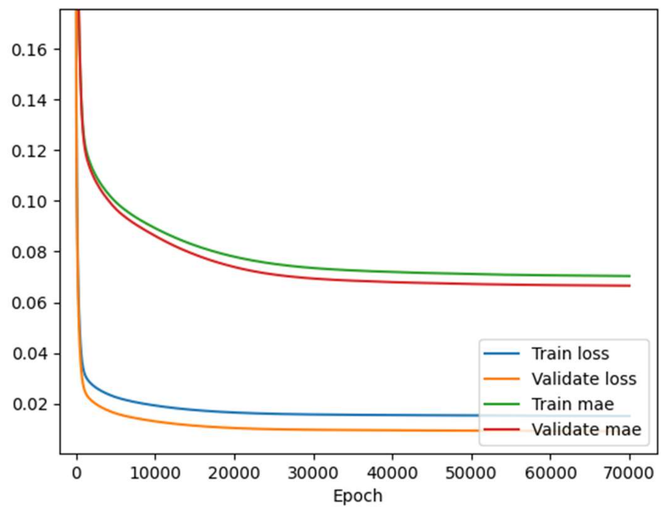


Figure 25: Cita single layer 36

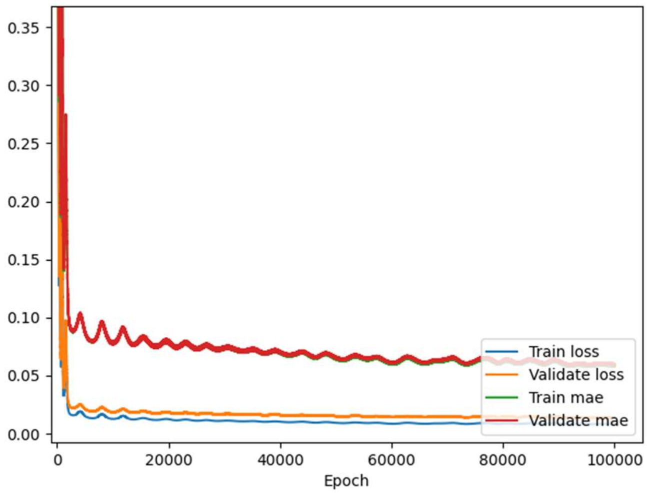


Figure 26: Delta one layer 36

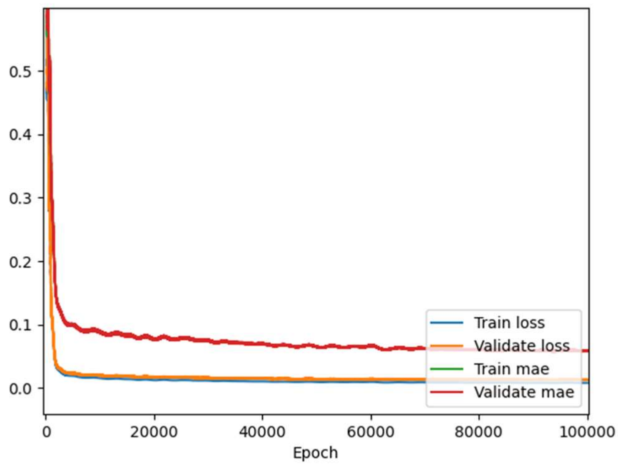


Figure 27: Delta single layer 72

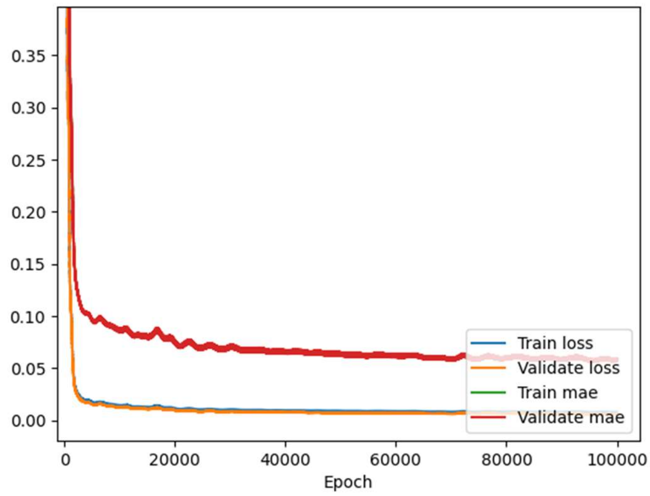


Figure 28: Delta single layer 128

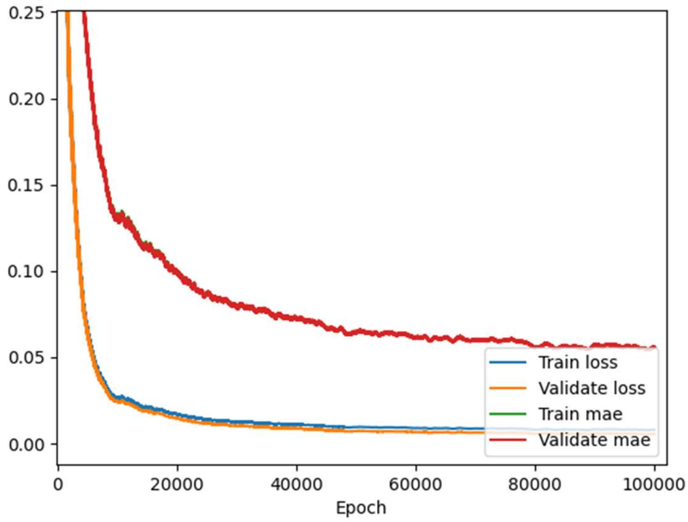


Figure 29: delta bi-layer 36 36

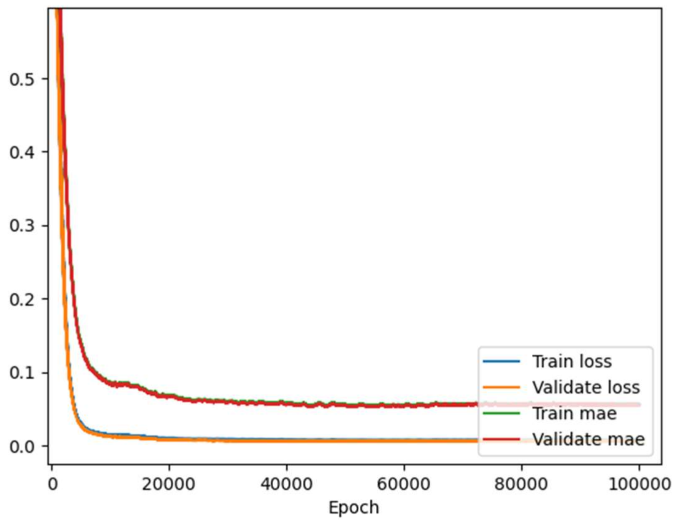


Figure 30: Delta tri-layer 128 256 128

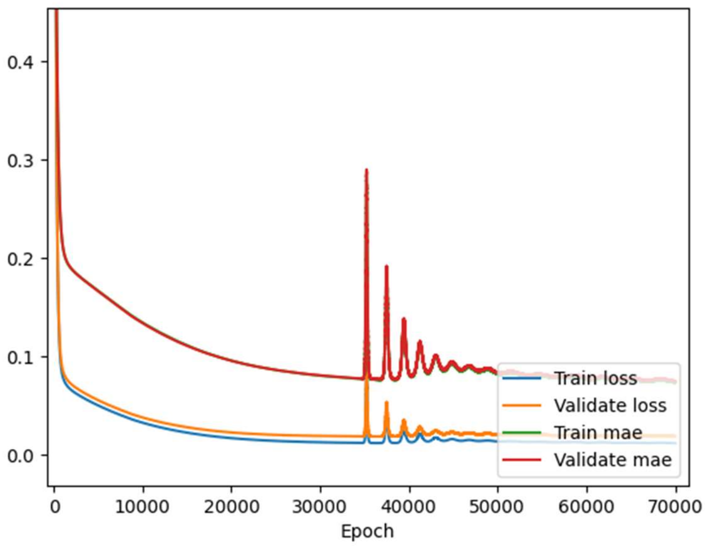


Figure 31: Alpha single layer 36

DATASET

https://drive.google.com/drive/folders/1UWo-fbiHtoGvtnNX8bOBrp2-YzMByVrA?usp=sharing